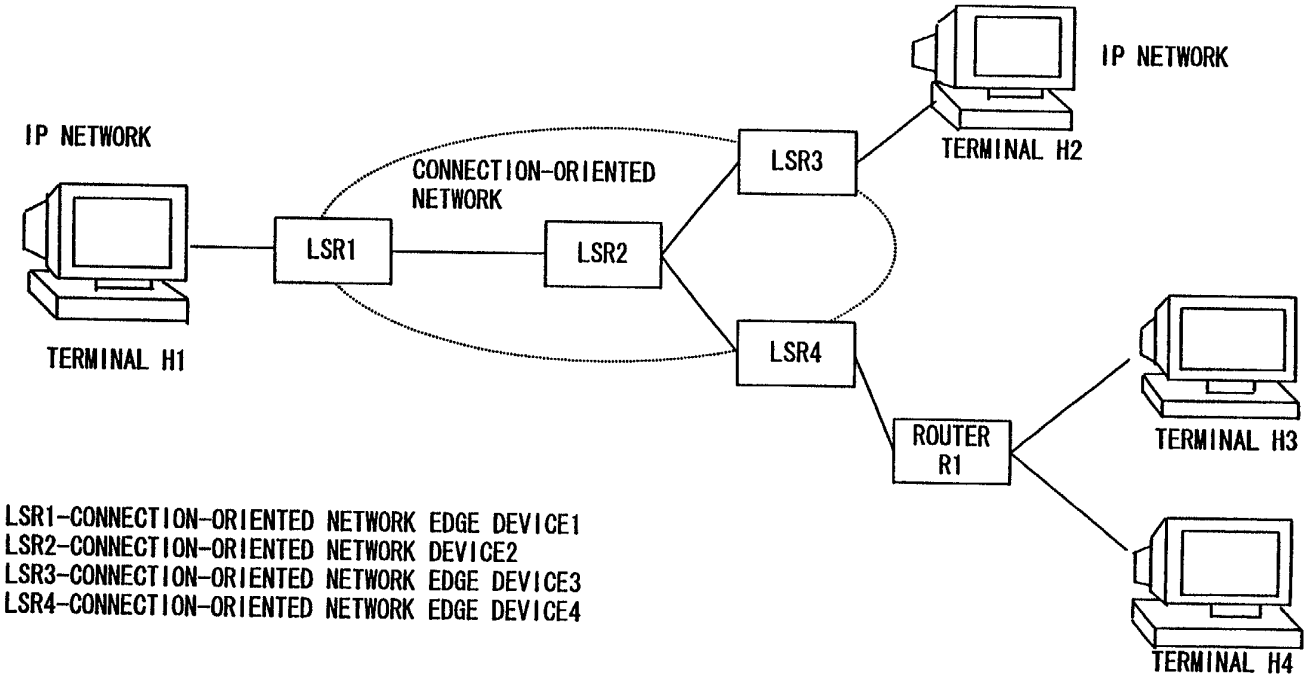

IP NETWORK
TERMINAL H1
LSR1
CONNECTION-ORIENTED NETWORK
LSR2
LSR3
LSR4
IP NETWORK
TERMINAL H2
ROUTER R1
TERMINAL H3
TERMINAL H4
LSR1-CONNECTION-ORIENTED NETWORK EDGE DEVICE1
LSR2-CONNECTION-ORIENTED NETWORK DEVICE2
LSR3-CONNECTION-ORIENTED NETWORK EDGE DEVICE3
LSR4-CONNECTION-ORIENTED NETWORK EDGE DEVICE4

FIG. 1

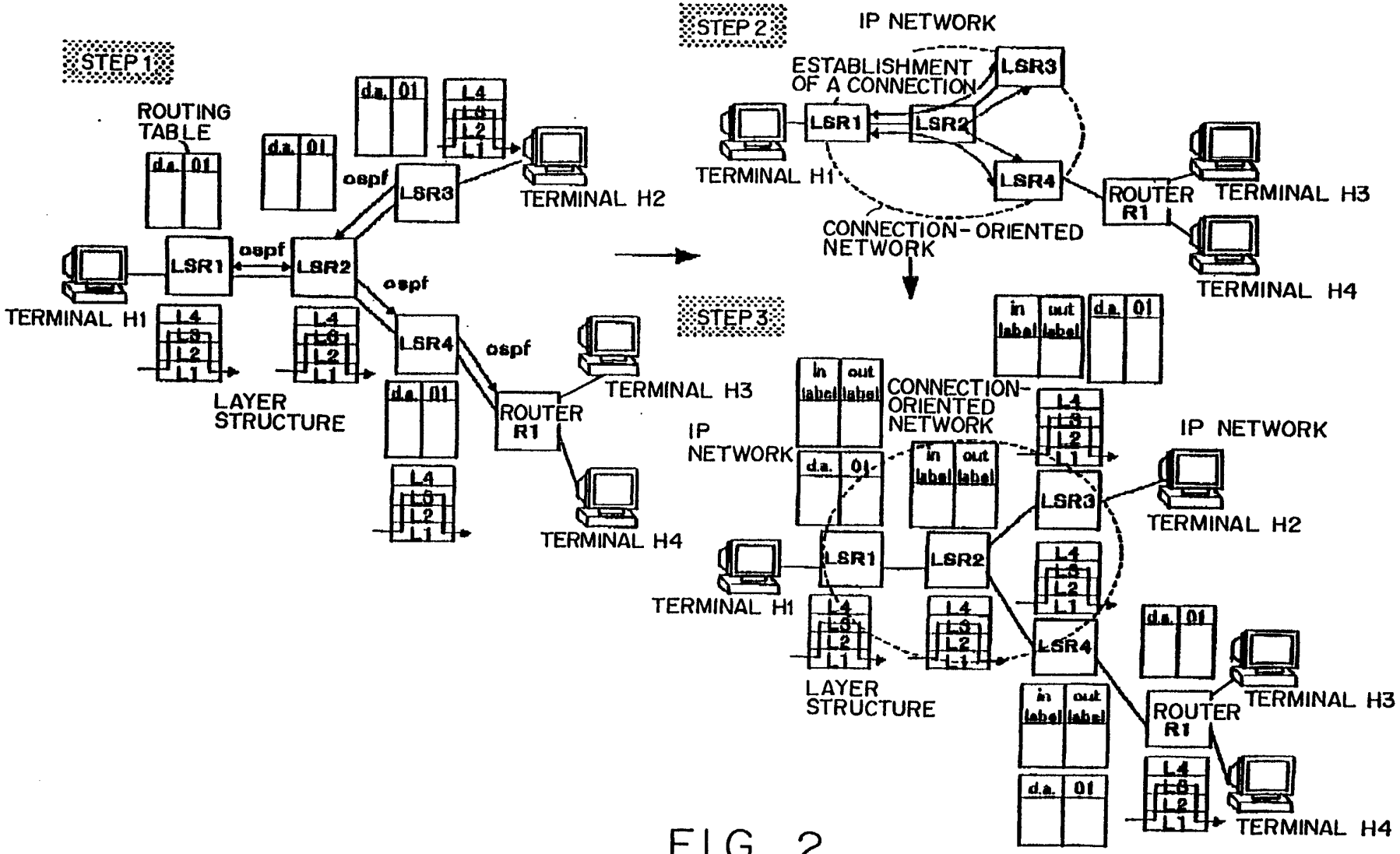

STEP 1
ROUTING TABLE
d.a. 01
ospf
LSR1
LSR2
LSR3
LSR4
TERMINAL H1
TERMINAL H2
TERMINAL H3
TERMINAL H4
LAYER STRUCTURE
ROUTER R1
STEP 2
IP NETWORK
ESTABLISHMENT OF A CONNECTION
CONNECTION-ORIENTED NETWORK
STEP 3
in label
out label
CONNECTION-ORIENTED NETWORK
IP NETWORK

FIG. 2

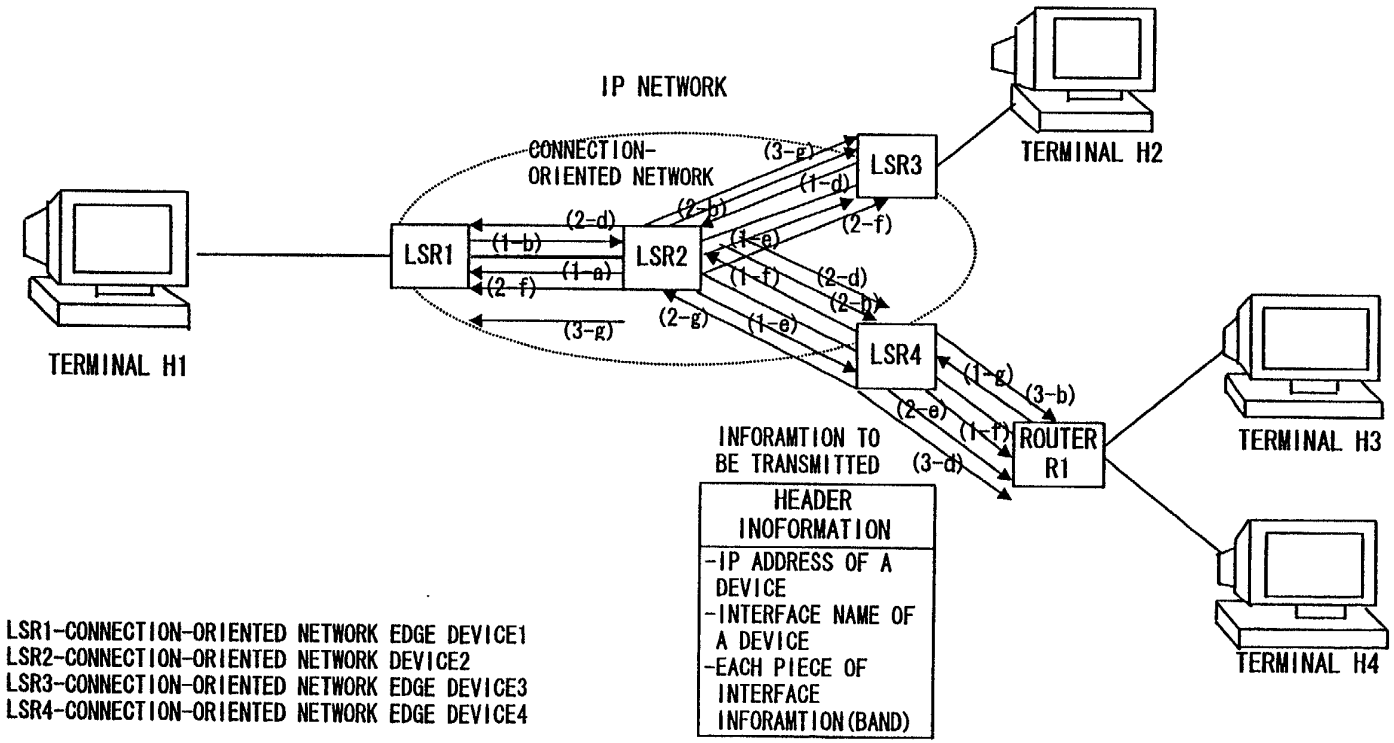

IP NETWORK
CONNECTION-
ORIENTED NETWORK
LSR1
LSR2
LSR3
LSR4
ROUTER
R1
TERMINAL H1
TERMINAL H2
TERMINAL H3
TERMINAL H4
(1-a)
(1-b)
(1-d)
(1-e)
(1-f)
(1-g)
(2-d)
(2-e)
(2-f)
(2-g)
(3-b)
(3-d)
(3-e)
(3-g)
INFORAMTION TO
BE TRANSMITTED
HEADER
INOFORMATION
-IP ADDRESS OF A
DEVICE
-INTERFACE NAME OF
A DEVICE
-EACH PIECE OF
INTERFACE
INFORAMTION(BAND)
LSR1-CONNECTION-ORIENTED NETWORK EDGE DEVICE1
LSR2-CONNECTION-ORIENTED NETWORK DEVICE2
LSR3-CONNECTION-ORIENTED NETWORK EDGE DEVICE3
LSR4-CONNECTION-ORIENTED NETWORK EDGE DEVICE4

FIG. 3

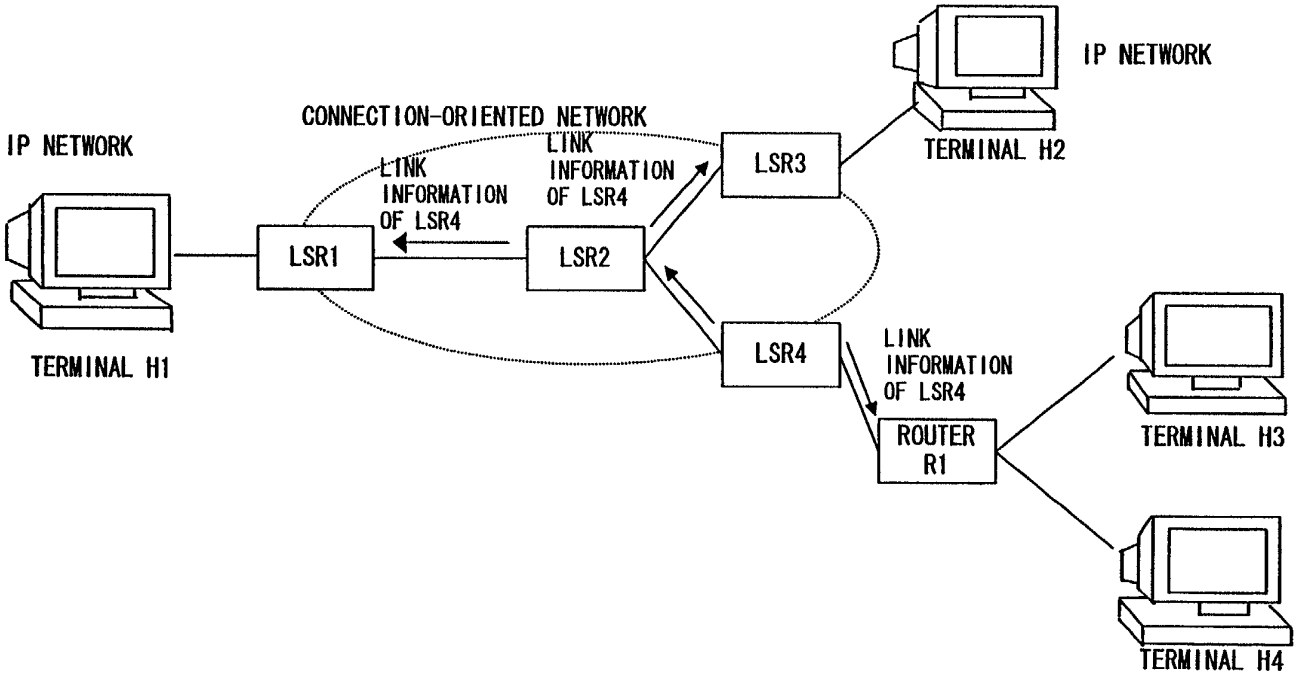

IP NETWORK
CONNECTION-ORIENTED NETWORK
IP NETWORK
LINK INFORMATION OF LSR4
LINK INFORMATION OF LSR4
LSR3
TERMINAL H2
LSR1
LSR2
TERMINAL H1
LSR4
LINK INFORMATION OF LSR4
ROUTER R1
TERMINAL H3
TERMINAL H4

FIG. 4

*	O	DC	EA	N/P	MC	E	*

THE OPTIONS FIELD

FIG. 5A

L	O	DC	EA	N/P	MC	E	*

THE OPTIONS FILED

FIG. 5B

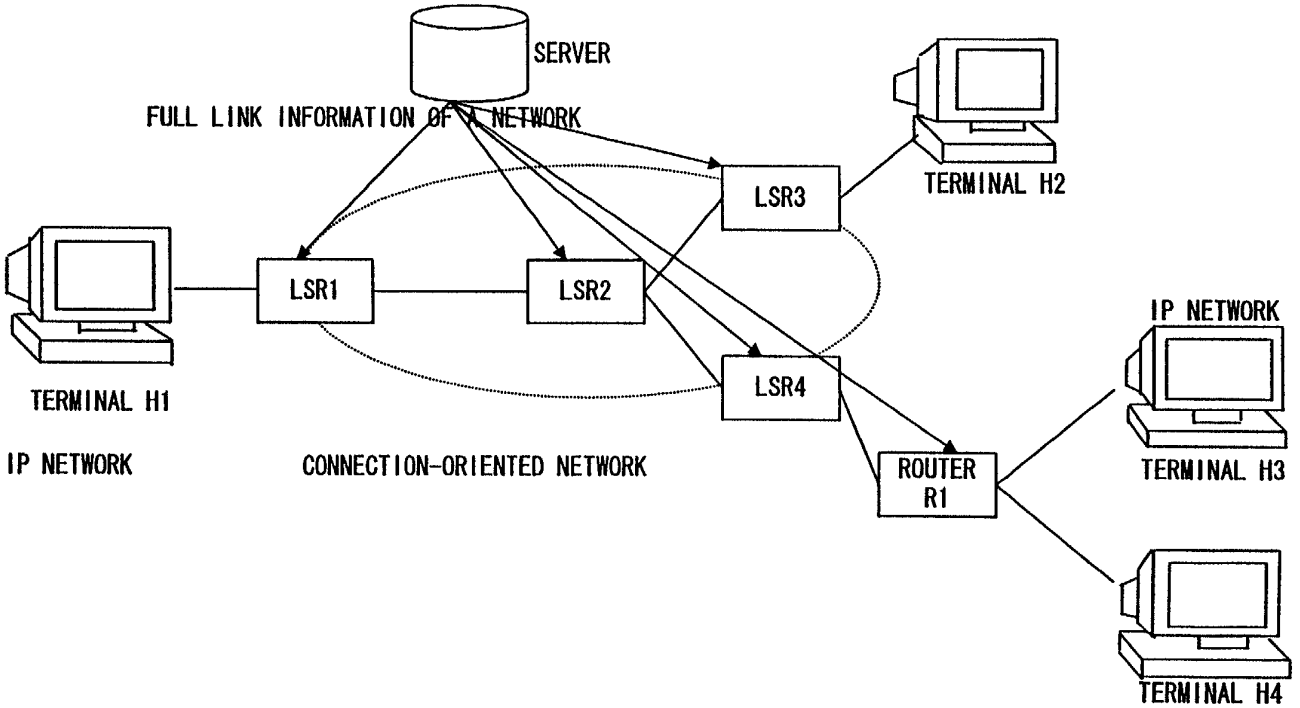

SERVER
FULL LINK INFORMATION OF A NETWORK
LSR3
TERMINAL H2
LSR1
LSR2
TERMINAL H1
LSR4
IP NETWORK
TERMINAL H3
IP NETWORK
CONNECTION-ORIENTED NETWORK
ROUTER R1
TERMINAL H4

FIG. 6

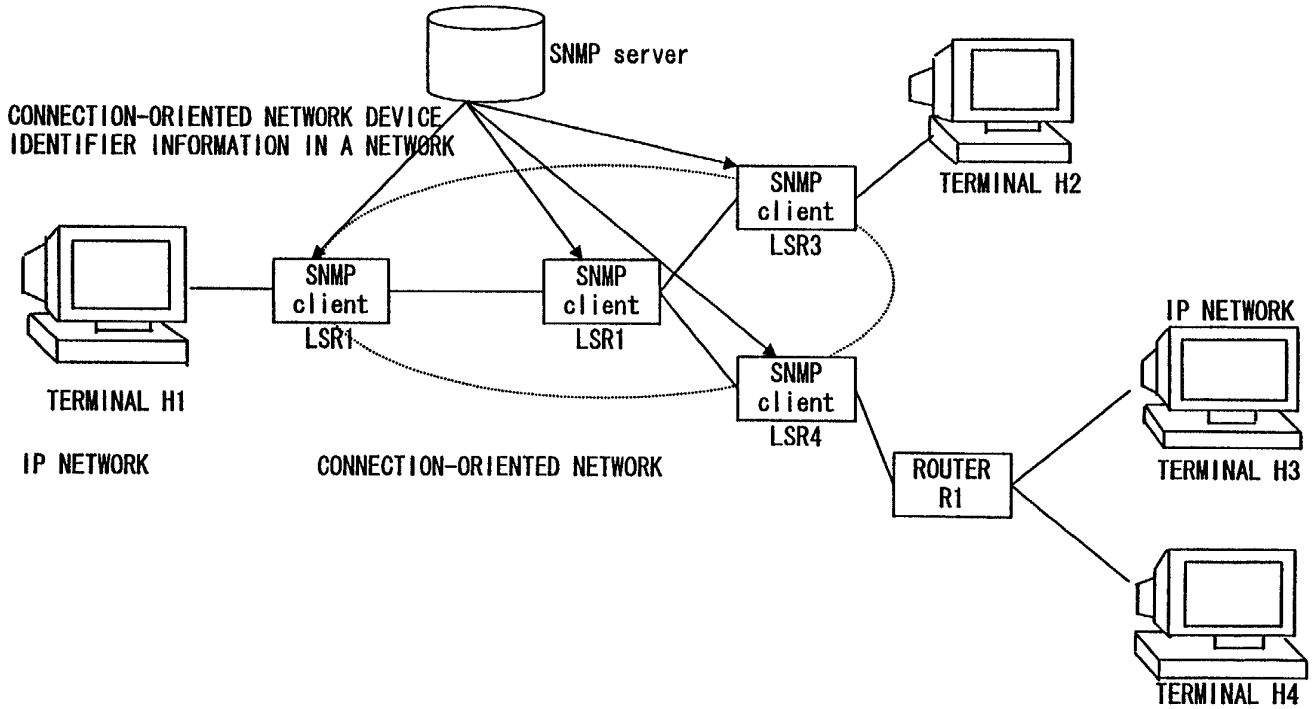

SNMP server
CONNECTION-ORIENTED NETWORK DEVICE IDENTIFIER INFORMATION IN A NETWORK
SNMP client LSR3
TERMINAL H2
SNMP client LSR1
SNMP client LSR1
TERMINAL H1
SNMP client LSR4
IP NETWORK
TERMINAL H3
IP NETWORK
CONNECTION-ORIENTED NETWORK
ROUTER R1
TERMINAL H4

FIG. 7

*	0	DC	EA	N/P	MC	E	R

THE OPTIONS FIELD

FIG. 8

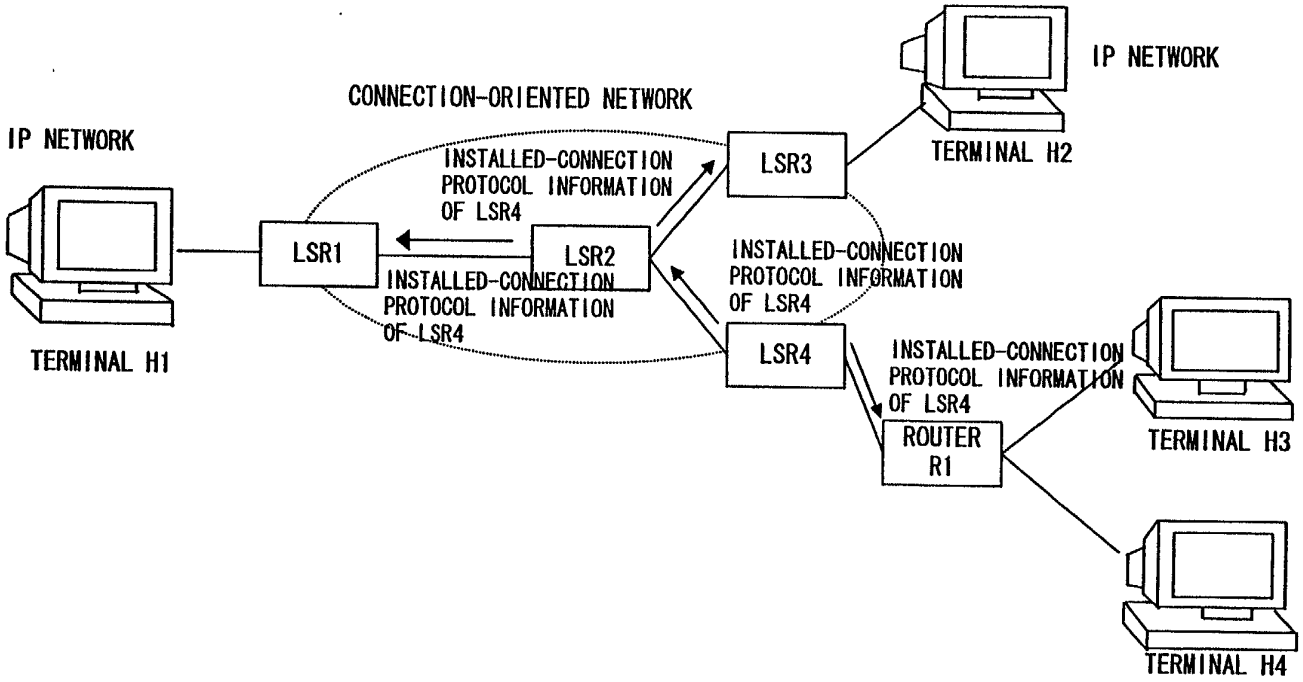

IP NETWORK
CONNECTION-ORIENTED NETWORK
IP NETWORK
TERMINAL H2
INSTALLED-CONNECTION PROTOCOL INFORMATION OF LSR4
LSR3
LSR1
LSR2
INSTALLED-CONNECTION PROTOCOL INFORMATION OF LSR4
INSTALLED-CONNECTION PROTOCOL INFORMATION OF LSR4
TERMINAL H1
LSR4
INSTALLED-CONNECTION PROTOCOL INFORMATION OF LSR4
ROUTER R1
TERMINAL H3
TERMINAL H4

FIG. 9

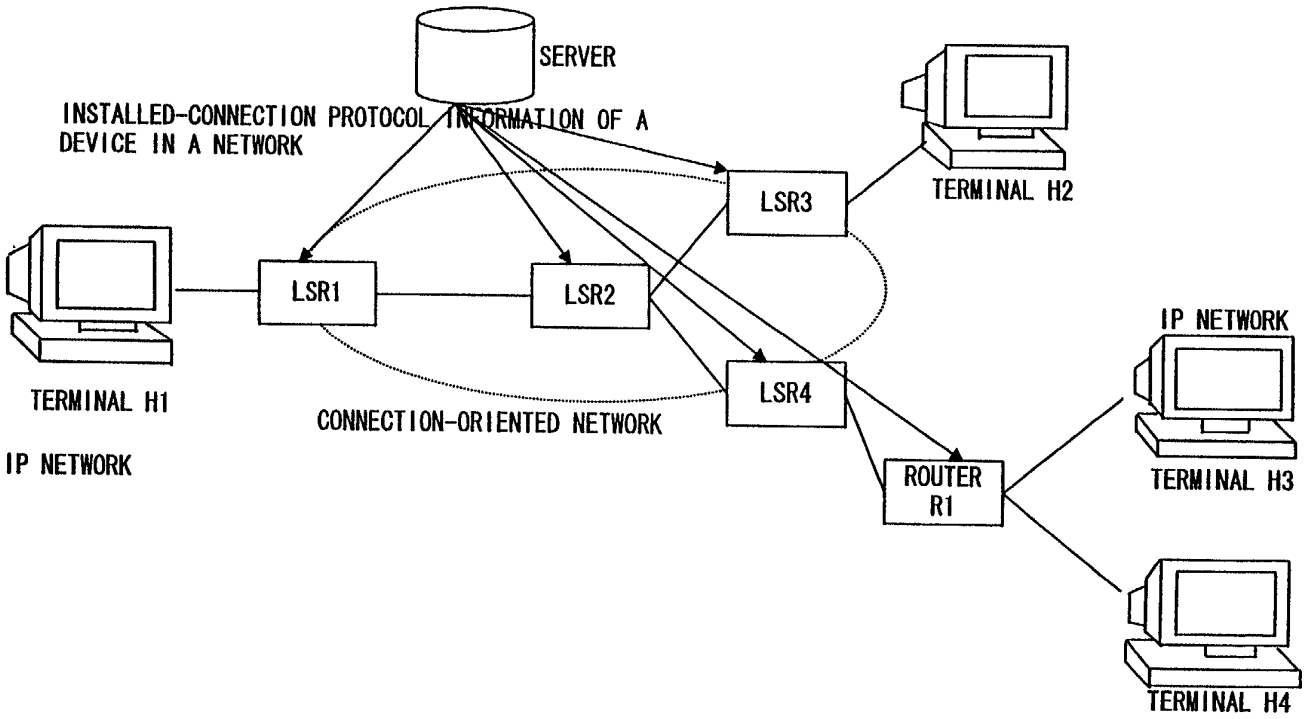

SERVER
INSTALLED-CONNECTION PROTOCOL INFORMATION OF A DEVICE IN A NETWORK
LSR3
TERMINAL H2
LSR1
LSR2
TERMINAL H1
IP NETWORK
CONNECTION-ORIENTED NETWORK
LSR4
IP NETWORK
ROUTER R1
TERMINAL H3
TERMINAL H4

FIG. 10

DEVICE(INTERFACE) ADDRESS	CONNECTION-ORIENTED NETWORK DEVICE IDENTIFIER	CONNECTION PROTOCOL IDENTIFIER
10.0.0.1	○	○
10.25.1.1	○	×
...	...	...

FIG. 11

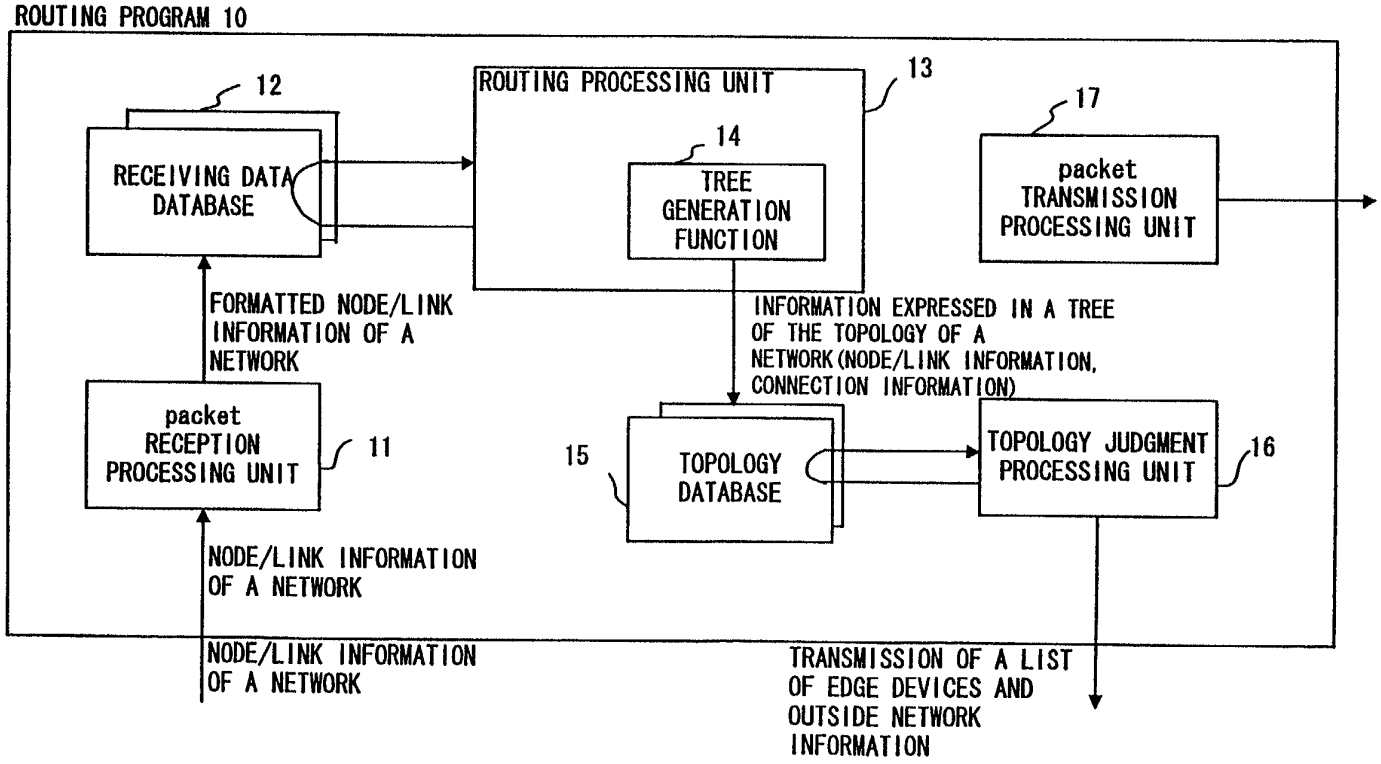

ROUTING PROGRAM 10
12
RECEIVING DATA DATABASE
ROUTING PROCESSING UNIT
13
14
TREE GENERATION FUNCTION
17
packet TRANSMISSION PROCESSING UNIT
FORMATTED NODE/LINK INFORMATION OF A NETWORK
INFORMATION EXPRESSED IN A TREE OF THE TOPOLOGY OF A NETWORK (NODE/LINK INFORMATION, CONNECTION INFORMATION)
packet RECEPTION PROCESSING UNIT
11
15
TOPOLOGY DATABASE
TOPOLOGY JUDGMENT PROCESSING UNIT
16
NODE/LINK INFORMATION OF A NETWORK
NODE/LINK INFORMATION OF A NETWORK
TRANSMISSION OF A LIST OF EDGE DEVICES AND OUTSIDE NETWORK INFORMATION

FIG. 12

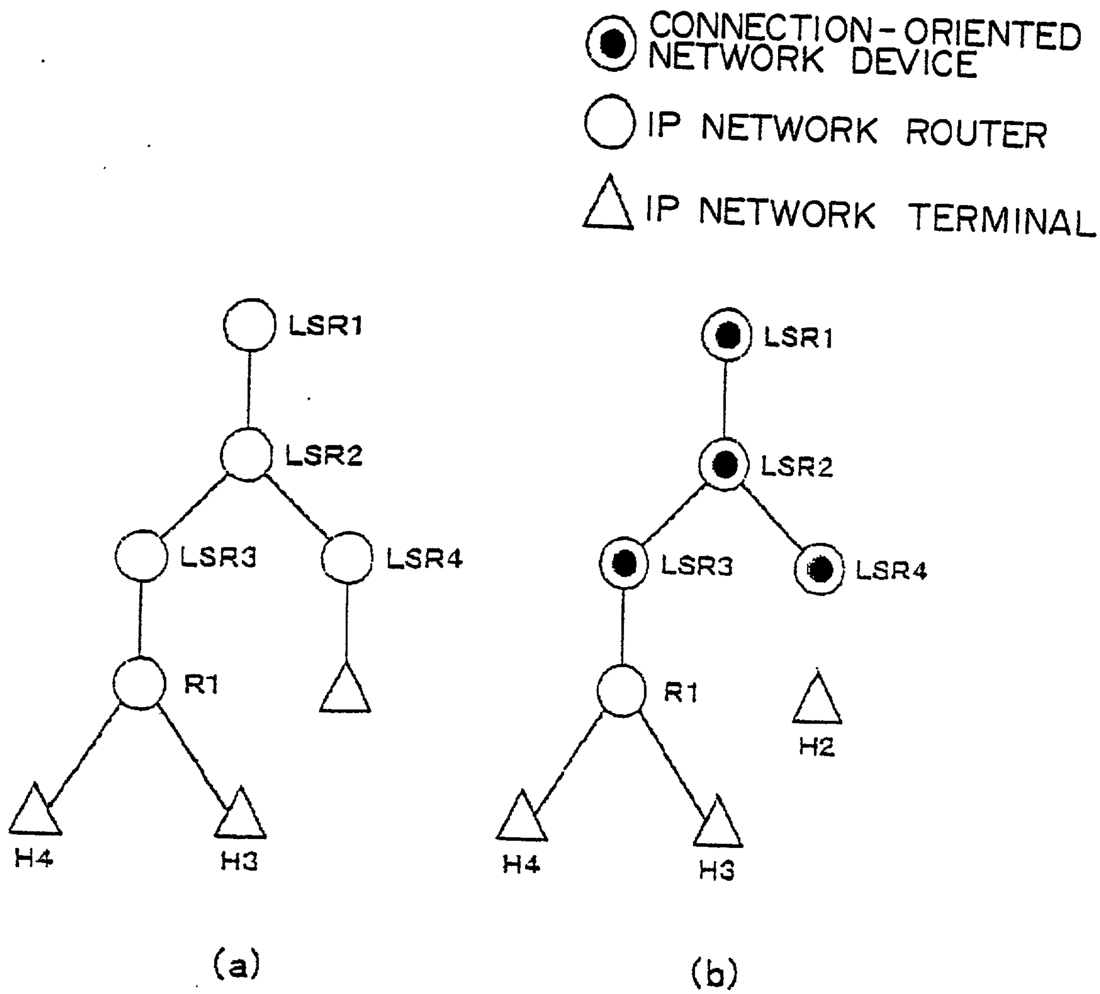

CONNECTION-ORIENTED NETWORK DEVICE
IP NETWORK ROUTER
IP NETWORK TERMINAL
LSR1
LSR2
LSR3
LSR4
R1
H4
H3
(a)
LSR1
LSR2
LSR3
LSR4
R1
H2
H4
H3
(b)

FIG. 13

```
/*initialization*/
DESIGNATES A SELF-NODE AS THE ROOT OF A TREE.
current_pointer=SELF NODE
/*spf rouine from here*/
spf_routine()
[
 while(1)[
        SEARCHES FOR LINK INFORMATION ADJACENT (RELATED) TO current_pointer
        IF(A NEW ENTRY IS DETECTED.)[
                ADDS THE NEW ENTRY TO THE TREE.
                IF(L/R BITS OF THE LINK INFORAMTION OF AN ENTRY "option header" BOTH ARE ON)[  .....(1)
                      THE DEVICE IS A VALID CONNECTION-ORIENTED NETWORK DEVICE
                      (THE INFORMATION IS INTERNALLY STORED).
                ]
                current_pointer=NODE OF A NEW ENTRY
        IF(THERE IS NO ENTRY.)[
                current_pointer=ONE-RANK HIGHER NODE
                IF(current_pointer=null)[
                        rrturn(0)+ /*THE SEARCH IS COMPLETED. EXITS THE ALGORITHM. */
                ]
        ]
  ]
]
```

FIG. 14

```
/*initialization*/
int traced []           /*CHECKS WHETHER ALL POSITIONS ARE CHEDCKED (CHECKED=1,
UNCHECKED=0)*/
int *current_pointer=LSR1  /*INITIALIZES THE CURRENT POSITION IN THE TREE.*/
char edge_entry []      /*EDGE DEVICE ENTRY*/
int edge_entry_number /*TOTAL NUMBER OF EDGE DEVICE ENTRIES*/

/*SEARCH ROUTINE*/
search()[
  while(1)[
        VIEWS THE CHILD OF A DEVICE POINTED TO BY current_pointer.
        IF(traced[child]=0){
                current_pointer=child; /*MOVES TO AN UNCHECKED CHILD*/
                traced[child]=1;
        }

  ELSE IF(ALL CHILDREN ARE CHECKED.){
                current_pointer=parent /*MOVES TO THE PARENT*/
                IF(parent=null){
                      break;          /*THE SEARCH IS COMPLETED.*/
                }
                 continue;
     }
     /*JUDGES WHETHER IT IS AN EDGE DEVICE.*/
     IF(L BIT POINTED TO BY current_pointer is zero.){
             edge_entry[edge_entry_number]=parent DEVICE;
             ++edge_entry_number;
             current_pointer=parent;
     }
  }
  return(0);
}
```

F I G. 1 5

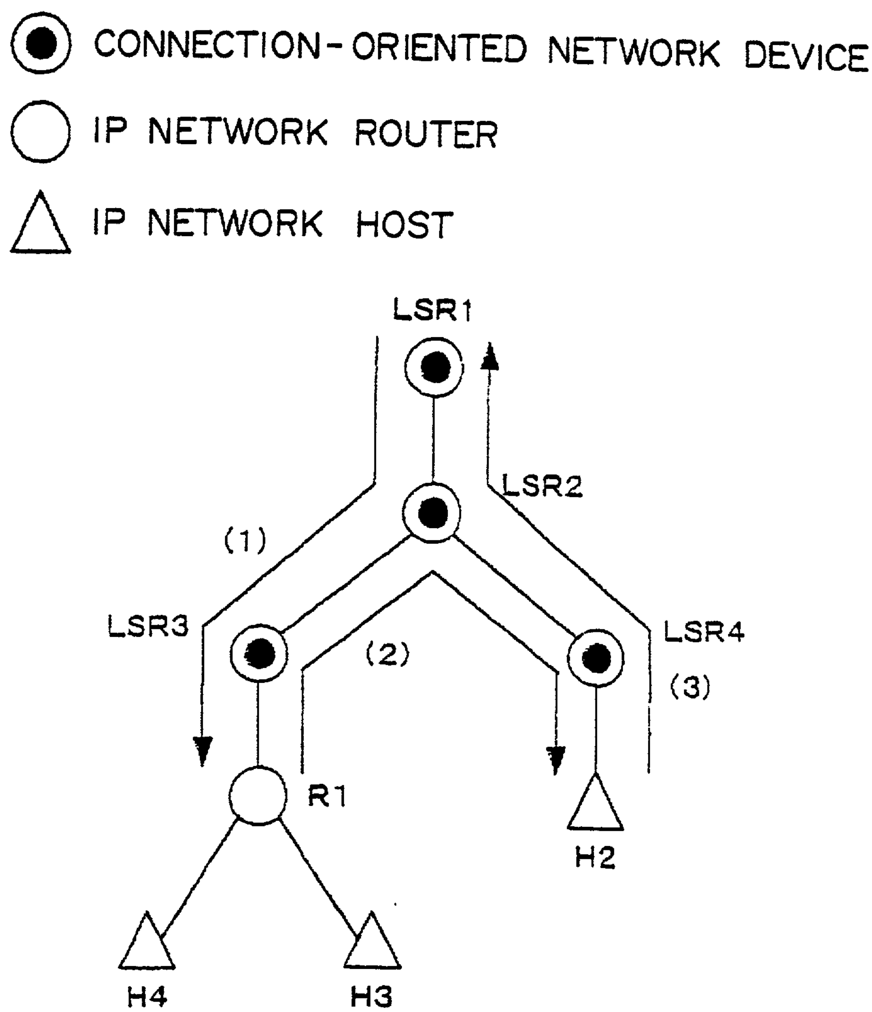

CONNECTION-ORIENTED NETWORK DEVICE
IP NETWORK ROUTER
IP NETWORK HOST
LSR1
LSR2
(1)
LSR3
(2)
LSR4
(3)
R1
H2
H4
H3

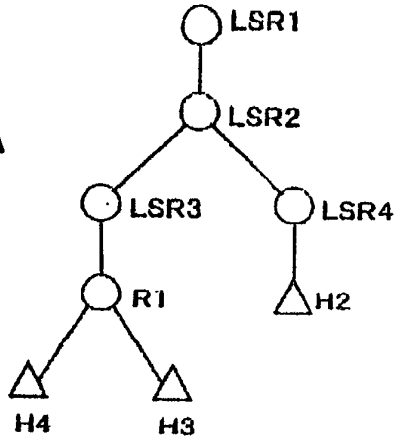
FIG. 16

EDGE DEVICE ENTRIES STORED IN LSR1

EDGE DEVICE
LSR3
LSR4

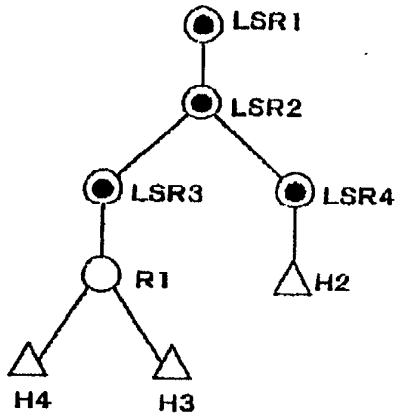
FIG. 17

FIG. 18A



destination address(d.a.)	next hop outgoint Interface(OI)

FIG. 18B



destination address(d.a.)	egress LSR IP address	next hop (outgoint Interface(OI)

```
/*initialization*/
int traced []        /*CHECKS WHETHER ALL POSITIONS ARE CHECKED (CHECKED=1, UNCHECKED=0)*/
int *current_pointer=LSR1 /*INITIALIZES THE CURRENT POSITION IN THE TREE.*/
char edge_entry []    /*EDGE DEVICE ENTRY*/
int edge_entry_number /*TOTAL NUMBER OF EDGE DEVIE ENTRIES*/
/*SERCH ROUTINE*/
search(){
  while(1)[
          VIEWS THE child OF A DEVICE POINTED TO BY current_pointer.
          IF(traced[child]=0){
                  current_pointer=child; /*MOVES TO AN UNCHECKED CHILD.*/
                  traced[child]=1:
          }
  ELSE IF(ALL CHILDREN ARE CHECKED.){
                  current_pointer=parent /*MOVES TO THE parent/*
                  IF(parent=null){
                        break;          /*THE SEARCH IS COMPLETED.*/
                  ]
                  continue;
          }

          /*JUDGES WHETHER IT IS AN EDGE DEVICE.*/
          IF(L BIT POINTED TO BY current_pointer IS ZERO.){
                    edge_entry[edge_entry_number]=parent DEVICE;
                    ++edge_entry number;
          }
          /*ADDS AN ENTRY OF A NETWORK CONNECTED TO AN EDGE DEVICE.*/
          IF(L BIT POINTED TO BY current_pointer IS ZERO.){
                    RELATES THE ip address POINTED TO BY curren_pointer TO
                    edge_entry[edge_entry_number](ENTRY ADDITION);
          ]
  }
  return(0);
]
```

FIG. 19

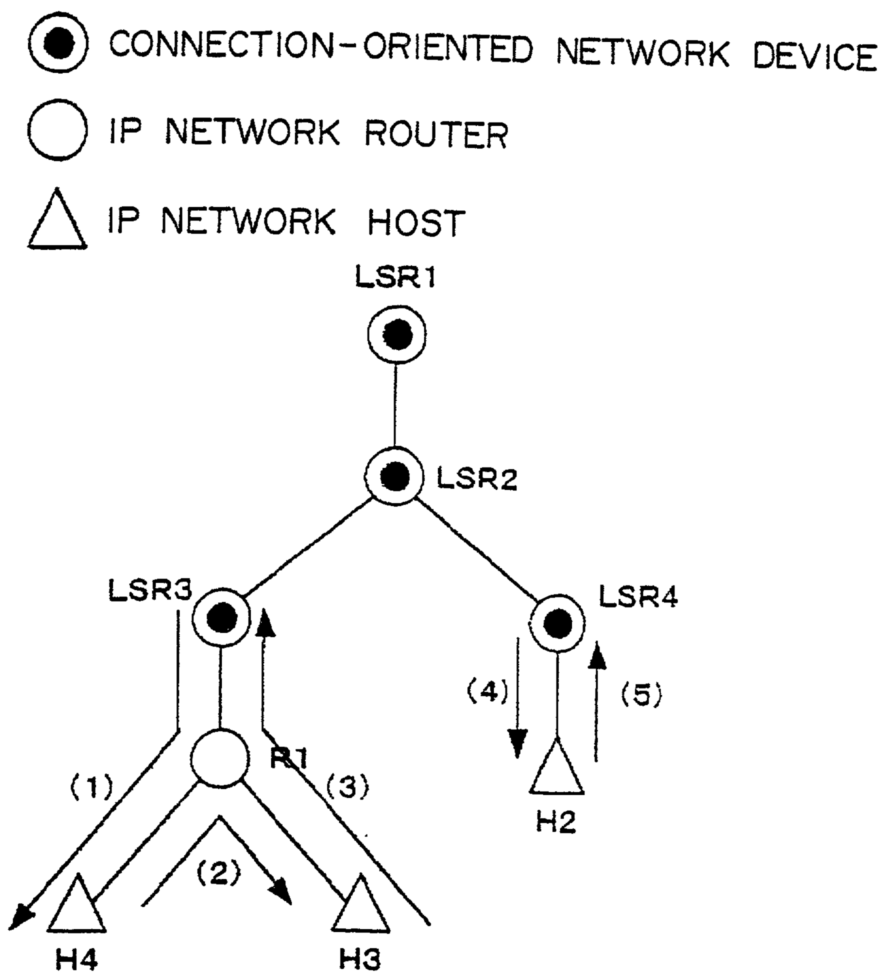

CONNECTION-ORIENTED NETWORK DEVICE
IP NETWORK ROUTER
IP NETWORK HOST
LSR1
LSR2
LSR3
LSR4
R1
(1)
(2)
(3)
(4)
(5)
H2
H3
H4

FIG. 20

ENTRY OF EDGE DEVICE/OUTSIDE NETWORK INFORMATION STORED IN LSR1

EDGE DEVICE	OUTSIDE NETWORK
LSR3	R1
LSR3	H4
LSR3	H3
LSR4	H2

FIG. 21

routing_table_request object
OBJECT INSERTED IN A PATH MESSAGE. IF SENDER(ENTRANCE EDGE DEVICE) WANTS TO OBTAIN THE ROUTING TABLE OF AN EXIST EDGE DEVICE, routing_table_request object IS INCLUDED IN THE PATH MESSAGE.

routing_table object
ON RECEIPT OF THE PATH MESSAGE, INCLUDING THE routing_table_request object, AN EXIT EDGE DEVICE RETURNS AN RESV MESSAGE, INCLUDING THE routing_table object, TO THE SENDER. THE file OF THE routing table IS COPIED INTO THE routing_table object AND IS TRANSMITTED.

FIG. 22

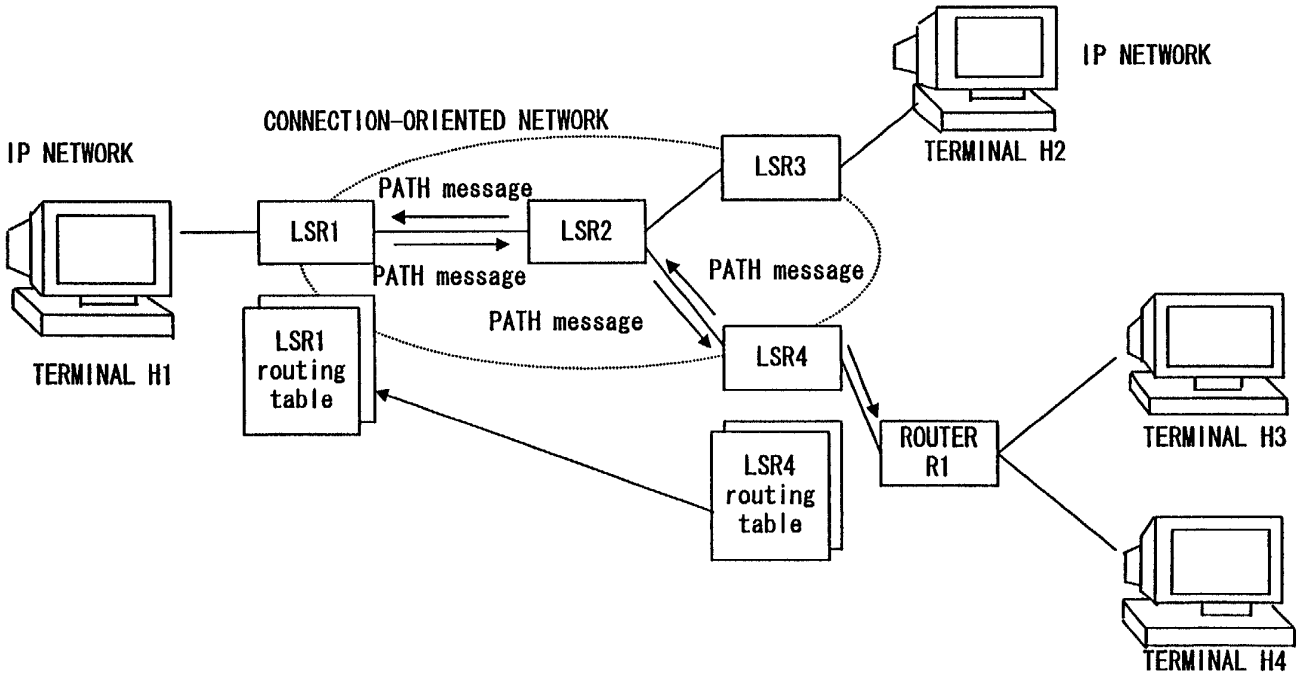

IP NETWORK
CONNECTION-ORIENTED NETWORK
IP NETWORK
LSR3
TERMINAL H2
PATH message
LSR1
LSR2
PATH message
PATH message
PATH message
LSR1
routing
table
LSR4
TERMINAL H1
TERMINAL H3
ROUTER
R1
LSR4
routing
table
TERMINAL H4

FIG. 23

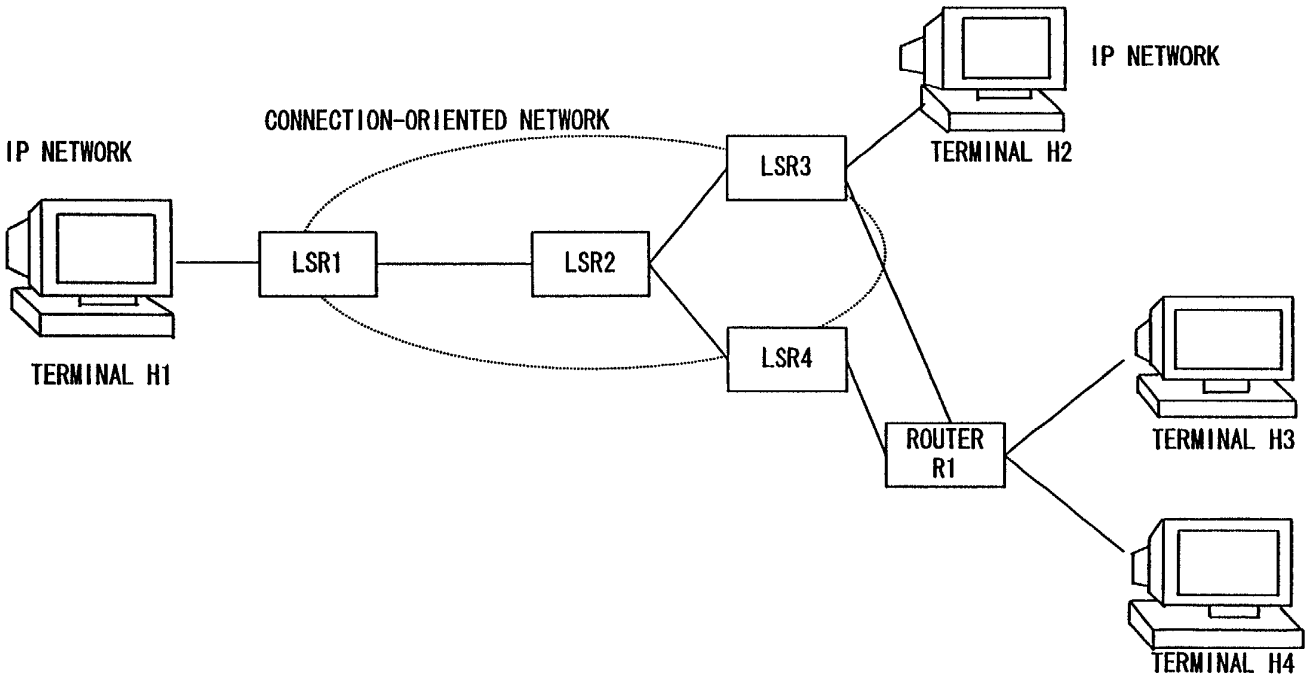

IP NETWORK
CONNECTION-ORIENTED NETWORK
IP NETWORK
LSR1
LSR2
LSR3
LSR4
TERMINAL H1
TERMINAL H2
ROUTER
R1
TERMINAL H3
TERMINAL H4

FIG. 24

FEC					label
d. a.	s. a.	d. p.	s. p.	proto	
10. 0. 0. 1	. . . .	1000	. . . .	. . . .	50
10. 0. 0. 1	. . . .	1050	. . . .	. . . .	60
20. 0. 0. 0	. . . .	. . . .	. . . .	. . . .	100

d. a. =destination IP address, s. a. =source IP address
d. p. =destination port, s. p. =source port
proto=protocol ID, ・・・=no-designation

FIG. 25

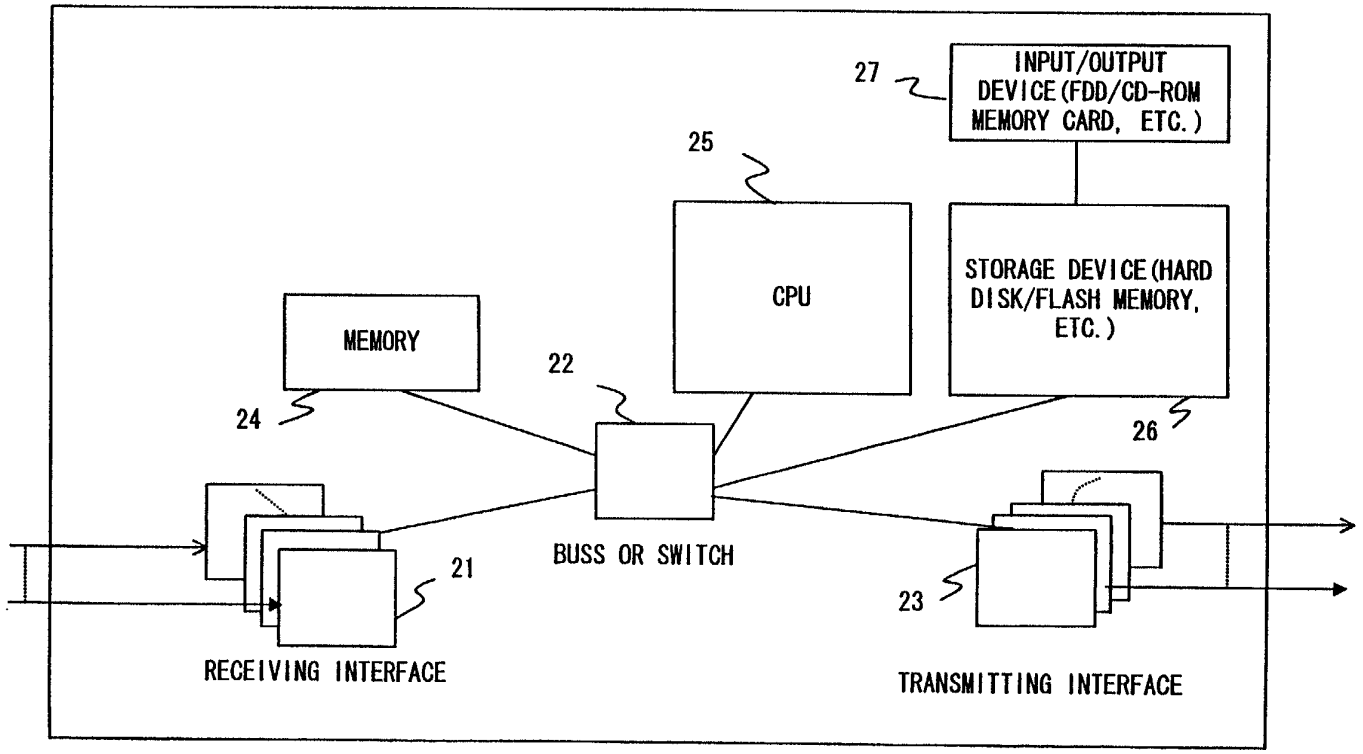

27
INPUT/OUTPUT DEVICE(FDD/CD-ROM MEMORY CARD, ETC.)
25
CPU
STORAGE DEVICE(HARD DISK/FLASH MEMORY, ETC.)
MEMORY
22
24
26
BUSS OR SWITCH
21
23
RECEIVING INTERFACE
TRANSMITTING INTERFACE

FIG. 26